REMARKS

Applicants request a clarification from the Examiner as to the claims under examination presently in this application. The Examiner states in the Office Action Summary that "Claims 7, 11 - 13 and 59 are pending in the application." In the January 13, 2005 Official Action, the Examiner stated that *"...claims 7, 9 - 13, 40 and 59 are allowable for the features of splicing a plurality of secure communication protocols of different protocol suites into the agent, wherein the step of splicing a plurality of secure communication protocols is a security protocol of a WAP to that of an IP...Cashman does not disclose these protocols..."* The Examiner is requested to advise as top the status of Claims 9, 10 and 40.

The Examiner is respectfully requested to reconsider the rejection of claims 7, 11 - 13, and 59 under 35 U.S.C. § 103 (a) as being unpatentable over Cashman, et al. (U.S. Patent 6,209,087) in view of Lincke (U.S. Patent 6,397,259).

The present invention relates to secure proxying for computing devices. The invention is directed toward network security protocols which are used to insure privacy and integrity of communication on an open public network. These protocols are intended to achieve end-to-end security guarantees such that the communication is private to the entities that establish the parameters of the secure communication channel.

Claim 7 now defines the use of a secure coprocessor which is used to achieve end to end security guarantees in the protocol translation between client and server which assures that the proxy cannot tamper with the functioning of said agent and view unencrypted communication between said client and said server, said agent being a software program or hardware logic operating within the confines of said secure coprocessor.

Cashman et al. describe a method which uses a coprocessor to implement elements of the protocol translation process between client and server. The Examiner does not accept that a coprocessor (which is under the control of the proxy) is differently trusted from a secure

coprocessor in our claim which enforces a very strong trust model. Claim 7 states that the protocols that the secure coprocessor will splice are the security protocols of WAP and SSL/TLS.

In Cashman's system, the proxy is trusted to do the aforementioned protocol translation, and the coprocessor is used merely as perfomance enhancing means. The Examiner unequivocally states in the Official Action cited above, that Cashman does not disclose splicing a plurality of secure communication protocols of different protocol suites into the agent wherein the step of splicing a plurality of secure communication protocols is a security protocol of a WAP to that of an IP device.

Note that in Cashman, the proxy can, and does tamper with what the coprocessor does. In Cashman, the proxy directly controls the coprocessor. Applicants re-emphasize that there is no end to end security guarantee being maintained by the protocol translation process of Cashman as is the case in the present invention.

The Examiner cites Lincke, et al. to supplement the Cashman reference to supply the teaching that Cashman admittedly is lacking.

Lincke discloses an improved system and method for handheld device to access Internet information over relative low bandwidth networks. Lincke is directed toward a communications system which includes the wireless communications device, a server, and a source of data. The server acts as a proxy server. Typical sources of data are a web server or a mail server. In reviewing the Lincke, et al. patent, it is not clear that the disclosure addresses the issue of splicing a plurality of secure communication protocols which is a security protocol of a WAP to that of an IP device. Primarily the Lincke patent seems to be about optimizing the number of messages sent to a wireless client. Note that in Cashman's scheme the contents are available unencrypted to any agent/process on the proxy.

The objective Lincke, et al. sought to emphasize was to consider wireless networks, such as those provided for two-way pagers and other wireless packet data networks, which provide

wider coverage and lower cost than competing networks. These wireless networks typically have relatively low performance however. A single packet of 400 bytes can take eight seconds just to travel to the Internet and back when the system is lightly loaded. With such a low throughput, it could easily take minutes to download even a small web page using standard browser technology. The wireless communications system therefore employs novel methods for reducing the amount of traffic sent over the wireless link for web access.

Lincke, et al. wanted to provide the user with fast access to web content. Although the wireless communications device can access generic web content, because of the wireless communications device's limited screen size, most existing content will not be as visually appealing, will be harder to navigate, and may take longer to access than specially formatted content. Thus, significantly advantages are achieved with customized content. The web content can be formatted for the small screens of most handheld communications devices. This content will download relatively quickly (because of its small size). The formatted content can be created and published using the same tools used today for desktop web publishing (i.e. HTML tools and web servers) and could even be viewed using a standard desktop browser.

Applicants use the coprocessor in their invention to enforce a trust model between the client and the server. The secure coprocessor guarantees that no external entity can tamper with the functioning of the hardware logic or software programs. The use of the coprocessor in the present invention insures that end to end security is guaranteed. Again something that Cashman does not insure nor does Lincke even consider.

It is essential to note with respect to the present invention, neither the proxy nor any external entity can tamper with the functionality being implemented by the software programs or hardware logic functioning with the confines of the coprocessor. This is not found in Lincke, et al.

Applicants respectfully submit that the specificity of the Cashman and Lincke, et al. disclosures alone or in combination, do not disclose or even imply the method of providing secure communication of the present invention as presently claimed. In the rejection, the Examiner is picking and choosing elements to the exclusion of what the references as a whole teach to one skilled in the art.

In order to analyze the propriety of the Examiner's obviousness rejections in this case, a review of the pertinent applicable law relating to 35 U.S.C. § 103 is warranted. The Examiner has applied the Cashman and Lincke, et al. references using selective combinations to render obvious the invention.

The Court of Appeals for the Federal Circuit has set guidelines governing such application of references. These guidelines are, as stated are found in Interconnect Planning Corp. v. Feil, 774 F.2d 1132, 1143, 227 USPQ, 543, 551:

> When prior art references require selective combination by the court to render obvious a subsequent invention, there must be some reason for the combination other than hindsight gleaned from the invention itself.

A representative case relying upon this rule of law is Uniroyal, Inc. v. Rudkin-Wiley Corp., 837 F.2d 1044, 5 USPQ 2d 1434 (Fed. Cir. 1988). The district court in Uniroyal found that a combination of various features from a plurality of prior art references suggested the claimed invention of the patent in suit. The Federal Circuit in its decision found that the district court did not show, however, that there was any teaching or suggestion in any of the references, or in the prior art as a whole, that would lead one with ordinary skill in the art to make the combination. The Federal Circuit opined:

> Something in the prior art as a whole must suggest the desirability, and thus the obviousness, of making the combination. [837 F.2d at 1051, 5 USPQ 2d at 1438, citing Lindemann, 730 F.2d 1452, 221 USPQ 481, 488 (Fed. Cir. 1984).]

Applicants respectfully submit that there is no basis for the combination of the Cashman and Lincke, et al. references cited by the Examiner. The Examiner has selected elements from these references for the sake of showing the individual elements claimed without regard to the total teaching of the references. As noted, the Examiner is improperly picking and choosing. The rejections are a piecemeal construction of the invention. Such piecemeal reconstruction of the prior art patents in light of the instant disclosure is contrary to the requirements of 35 U.S.C. § 103.

> The ever present question in cases within the ambit of 35 U.S.C. § 103 is whether the subject matter as a whole would have been obvious to one of ordinary skill in the art following the teachings of the prior art at the time the invention was made. It is impermissible within the framework of Section 103 to pick and choose from any one reference only so much of it as will support a given position, to the exclusion of other parts necessary to the full appreciation of what such reference fairly suggests to one of ordinary skill in the art. (Emphasis in original) In re Wesslau 147 U.S.P.Q. 391, 393 (CCPA 1965)

This holding succinctly summarizes the Examiner's application of references in this case because she did in fact pick and choose so much of the Cashman and Lincke, et al. disclosures to support her position and did not cover completely in the Office Action the full scope of what these varied disclosure references fairly suggest to one skilled in the art.

As noted above, Lincke, et al. desire to provide the user with fast access to web content. Cashman teaches against Lincke's objective stating at Column 3, lines 6 - 8 that *"A CPU executing a program to compress and encrypt data must process data fast enough to fully utilize available data communications bandwidth. Fast processors are expensive and and increase the cost of data communications devices."* Lincke, et al. state that "*A goal of the invention is to provide the user with fast access to web content.*" The teaching are diametrically opposite. There is no basis to combine the references as has been done in the Official Action, based upon their respective teachings and objectives. The Examiner has already stated on the record that the Claims are patentable over Cashman. The Lincke, et al. disclosure is a non-analogous art based upon its teaching. There is no proper basis to combine the references.

Further, the Federal Circuit has stated that the Patent Office bears the burden of establishing obviousness, and that this burden can only be satisfied by showing some objective teaching in the prior art or that knowledge generally available to one of ordinary skill in the art would lead that individual to combine the relevant teachings of the reference.

> Obviousness is tested by "what the combined teachings of the references would have suggested to those of ordinary skill in the art." In re Keller, 642 F.2d 413, 425, 208 USPQ 871, 881 (CCPA 1981). But it "cannot be established by combining the teachings of the prior art to produce the claimed invention, absent some teaching or suggestion supporting the combination." ACS Hosp. Sys., 732 F.2d at 1577, 221 USPQ at 933. [837 F.2d at 1075, 5 USPQ 2d at 1599.]

The Court concluded its discussion of this issue by stating that teachings or references can be combined only if there is some suggestion or incentive to do so.

In the present case, the skilled artisan viewing the Cashman and Lincke, et al. references would not be inclined to combine the be directed toward a totally different system than is called for in the present invention. There is no teaching in Lincke, et al. directed toward Applicants' objective of use a secure coprocessor to perform protocol translation in a manner that preserves the end to end trust model between the client and server. There is no mention of security and one cannot assume that combining the entire teaching of Lincke, et al. with Cashman would not compromise the Cashman system. The skilled artisan must consider the entire teaching of Lincke, et al. before combining it with Cashman. Applicants' Claims define a "secure coprocessor" which explicitly means tamper resistant/ tamper-proof and further means that the coprocessor is translating protocols while still maintaining the trust model between the client and server. Neither Cashman nor Lincke, et al. disclose those elements. The combination of references is improper.

If there are issues which could be resolved by a telephone conference, Applicants' attorney would be willing to speak with the Examiner concerning such matter(s) at a mutually convenient time. The Examiner is requested to contact Applicant's attorney by telephone at the number listed below.

Applicant has attempted to distinguish the invention as embodied in the amended claims over the prior art. In view of the arguments and modifications to the claims, allowance of this case is warranted. Such favorable action is respectfully solicited.

The Commissioner is requested to grant a one month extension of time within which to respond to the above-identified Official Action. A check in the amount of $110.00 to cover the one month extension fee is enclosed.

October 31, 2005

Respectfully submitted,

Thomas A. Beck
Reg. No. 20,816
26 Rockledge Lane
New Milford, CT 06776

I certify that this amendment is being mailed via the United States Postal Service postage prepaid on the date shown below addressed to:
Assistant Commissioner of Patents, P.O. Box 1450. Alexandria, VA 22313-1450

Signature _____ Date: October 31, 2005

Name: Thomas A. Beck

APPENDIX A
CLAIMS

1. (Cancelled) A method for achieving client to server end to end security guarantees, comprising:

providing a secure communication between a client and a server employing an untrusted proxy by means of:

employing said proxy between a said client and a said server to provide connection links between said client and said server;

embedding a secure coprocessor for use as an agent of said client and/or said server which assures that said proxy cannot tamper with the functioning of said agent, said agent being a software program or hardware logic operating within the confines of said coprocessor;

said proxy receiving a specific communication request from said client;

said coprocessor is located at the site of said proxy and: (a) acts as a converter between at least one protocol said client supports, and at least one other protocol supported by said server, (b) guarantees that an application embedded in said coprocessor performs to a degree of security proscribed by said client and/or said server;

said proxy forming an n-tuple for a specific communication;

said proxy forwarding said n-tuple to said coprocessor;

said coprocessor generating a response, including a directive to said n-tuple;

said coprocessor sending said response to said proxy and

said proxy implementing a directive; and

employing the respective security protocols of said at least one protocol and said at least one other protocol.

Claims 2 - 4 (Canceled)

Claim 5 (Canceled) A method as recited in claim 1 wherein the client is a pervasive computing device.

Claim 6 (Canceled) A method as recited in claim 5 further comprising the step of adapting content supplied by the client to fit constraints of the server and/or the connection links.

Claim 7 (Currently Amended) A method for providing secure communications on a network, the method comprising:
providing a secure communication between a client and a server employing an untrusted proxy by means of:

employing said proxy between said client and said server to provide connection links between said client and said server;

embedding a secure coprocessor for use as an agent of said client and/or said server which assures that said proxy cannot tamper with the functioning of said agent and view unencrypted communication between said client and said server, said agent being a software program or hardware logic operating within the confines of said secure coprocessor;

said proxy receiving a specific encrypted communication request from said client;

said coprocessor is located at the site of said proxy and: (a) acts as a converter between at least one protocol said client supports, and at least one other protocol supported by said server, (b) guarantees that an application embedded in said coprocessor performs to a degree of security proscribed by said client and/or said server;

said proxy forming an n-tuple for a specific communication;

said proxy forwarding said n-tuple to said coprocessor;

said coprocessor generating a response, including a directive to said n-tuple;

said coprocessor sending said response to said proxy and

said proxy implementing a directive; and

employing the respective security protocols of said at least one protocol and said at least one other protocol;

splicing a plurality of secure communication protocols of different protocol suites into the agent, wherein the step of splicing a plurality of secure communication protocols is a security protocol of a Wireless Application Protocol Suite (WAP) to that of an Internet Protocol (IP) device, said WAP being used by a pervasive computing device, and said agent performs at least one content adaptation function.

Claim 8 (Canceled) A method as recited in claim 7 wherein the step of splicing includes splicing a security protocol of a Wireless Application Protocol Suite (WAP) to that of an Internet Protocol (IP) device.

Claim 9 (Previously Presented) A method as recited in claim 7 wherein the Wireless Application Protocol suite is used by a pervasive computing device.

Claim 10 (Previously Presented) A method as recited in claim 9 further comprising the agent performing at least one content adaptation function.

Claim 11 (Previously presented) A method as recited in claim 10, wherein the step of performing includes maintaining communication privacy.

Claim 12 (Previously Presented) A method as recited in claim 10, further comprising maintaining a state of splicing process resulting from the step of splicing.

Claim 13 (Previously presented) A method as recited in claim 12, wherein the step of maintaining includes employing a storage device external to the proxy, and using cryptographic means to encrypt the state.

Claim 14 (Canceled) A method for providing network security to a network employing a proxy, the method comprising:

embedding a trusted application in a secure coprocessor located at the site of a proxy; and

delegating to a network infrastructure a task of enforcing a trust model.

Claim 15 (Canceled) A method as recited in claim 14, further comprising guaranteeing that the application is trusted to enforce th trust model between at least one server and a plurality of clients.

Claim 16 (Canceled) A method as recited in claim 14, further comprising assuring the tamper resistance of the application.

Claim 17 (Canceled) A method for secure communication between a client and a server employing an untrusted proxy; the method comprising:

embedding a coprocessor at the proxy;

the proxy receiving a specific communication request from a client;

the proxy forming an n-tuple for the specific communication;

the proxy forwarding the n-tuple to the coprocessor;

the coprocessor generating a response, including a directive, to the n-tuple;

the coprocessor sending the response to the proxy, and

the proxy implementing the directive.

Claim 18 (Canceled) A method of claim 17, wherein the coprocessor is a secure coprocessor.

Claim 19 (Canceled) A method of claim 17, wherein the step of receiving includes:

awaiting a connection request from a client;

creating an entry in a storage module for the client;

determining a sender of each received packet; and

retrieving a stored entry.

Claim 20 (Canceled) A method of claim 19, wherein the n-tuple includes a sender id, an entry from a storage module and the received packet.

Claim 21 (Canceled) A method of claim 17, wherein the client and the server can be either a sender or a receiver, and the step of generating includes employing a first protocol from the sender to the proxy and a second protocol from the proxy to the receiver and translating between the first and second protocols.

Claim 22 (Canceled) A method of claim 21, wherein the translating includes decrypting the received packet as specified by the security parameters negotiated as per the first protocol and encrypting the decrypted packet as specified by the security parameters of the second protocol.

Claim 23 (Canceled) A method of claim 21, wherein the translating includes modifying the received packet to meet constraints of the receiver and wherein the directive includes forwarding to the receiver the packet resulting from the step of modifying.

Claim 24 (Canceled) A method as recited in claim 23, further comprising aggregating a plurality of packets into a group of packets and performing content adaptation on the group of packets.

Claim 25 (Canceled) A method of claim 17, wherein the communication between the client and the proxy employ protocols specified by the Wireless Application Protocol suite (WAP).

Claim 26. (Canceled) A system to control security of a proxy interconnecting a client to a server, comprising:

providing a secure communication between a client and a server employing an untrusted proxy by means of:

employing said proxy between a said client and a said server to provide connection links between said client and said server;
embedding a secure coprocessor for use as an agent of said client and/or said server which assures that said proxy cannot tamper with the functioning of said agent, said agent being a software program or hardware logic operating within the confines of said coprocessor;
said proxy receiving a specific communication request from said client;

said coprocessor is located at the site of said proxy and: (a) acts as a converter between at least one protocol said client supports, and at least one other protocol supported by said server, (b) guarantees that an application embedded in said coprocessor performs to a

degree of security proscribed by said client and/or said server;

said proxy forming an n-tuple for a specific communication;

said proxy forwarding said n-tuple to said coprocessor;

said coprocessor generating a response, including a directive to said n-tuple;

said coprocessor sending said response to said proxy and

said proxy implementing a directive; and

employing the respective security protocols of said at least one protocol and said at least one other protocol;

said secure coprocessor, being used as an agent of the client and/or a server, said secure coprocessor being located at the site of said proxy ; said agent being a software program or hardware logic operating within the confines of said coprocessor and

an application embedded in said secure coprocessor which acts as a converter between at least one protocol said client supports and at least one other protocol supported by said server, wherein said secure coprocessor employs respective security protocols of said at least one protocol and said at least one other protocol; said secure coprocessor also assuring that said proxy cannot tamper with the functioning of said agent, and guaranteeing that an application embedded in said coprocessor performs to a degree of security proscribed by said client and/or said server.

Claims 27 - 29 (Canceled)

Claim 30 (Canceled) A system as recited in claim 26, wherein the application embedded in the coprocessor adapts content supplied by the server to fit constraints of the client and the connection links.

Claim 31 (Canceled) A system as recited in claim 30 wherein the application embedded in the coprocessor adapts content supplied by the client to fit constraints of the server and the connection links.

Claim 32 (Canceled) A system for providing network security to a network employing a proxy, the system comprising:

a secure coprocessor located at the site of a proxy; and

a trusted application embedded in the coprocessor wherein the coprocessor delegates the task of enforcing an arbitrary trust model to the application.

Claim 33 (Canceled) A system as recited in claim 32, wherein the coprocessor functions to guarantee that the application is trusted to enforce the trust model between at least one server and a plurality of clients.

Claim 34 (Canceled) A system as recited in claim 32, wherein the coprocessor functions to assure the tamper resistance of the application.

Claim 35. (Canceled) An article of manufacture comprising a computer usable medium having computer readable program code means embodied therein for achieving client to server end to end security guarantees, the computer readable program code means in said article of manufacture comprising computer readable program code means for causing a computer to effect: employing a proxy between a client and a server to provide connection links between said client and said server;

providing a secure communication between a client and a server employing an untrusted

proxy by means of:

embedding a secure coprocessor for use as an agent of said client and/or said server which assures that said proxy cannot tamper with the functioning of said agent, said agent being a software program or hardware logic operating within the confines of said coprocessor;

said proxy receiving a specific communication request from said client;

said coprocessor is located at the site of said proxy and: (a) acts as a converter between at least one protocol said client supports, and at least one other protocol supported by said server, (b) guarantees that an application embedded in said coprocessor performs to a degree of security proscribed by said client and/or said server;

said proxy forming an n-tuple for a specific communication;

said proxy forwarding said n-tuple to said coprocessor;

said coprocessor generating a response, including a directive to said n-tuple;

said coprocessor sending said response to said proxy and

said proxy implementing a directive; and

employing the respective security protocols of said at least one protocol and said at least one other protocol;
said coprocessor is located at said proxy and: (a) acts as a converter between at least one protocol said client supports, and at least one other protocol supported by said server, (b) and guarantees that an application embedded in said coprocessor performs to a degree

of security proscribed by said client and/or said server;

employing the respective security protocols of said at least one protocol and said at least one other protocol.

Claim 36 (Canceled) An article of manufacture as recited in claim 35, the computer readable code means in said article of manufacture further comprising computer readable program code means for causing a computer to effect the coprocessor assuring that the proxy can not tamper with the functioning of the agent.

Claim 37 (Canceled)

Claim 38. (Canceled) An article of manufacture comprising a computer usable medium having computer readable program code means embodied therein for achieving client to server end to end security guarantees, the computer readable program code means in said article of manufacture further comprising computer readable program code means for causing a computer to effect:

employing a proxy between a client and a server to provide connection links between said client and said server;

embedding a secure coprocessor for use as an agent of said client and/or said server which assures that said proxy cannot tamper with the functioning of said agent, said agent being a software program or hardware logic operating within the confines of said coprocessor;

said coprocessor is located at said proxy site and: (a) acts as a converter between at least one protocol said client supports, and at least one other protocol supported by said server, (b) adapts content supplied by said server to fit constraints of said client and/or connection

links.

employing the respective security protocols of said at least one protocol and said at least one other protocol.

Claim 39. (Canceled) An article of manufacture comprising a computer usable medium having computer readable program code means embodied therein for achieving client to server end to end security guarantees, the computer readable program code means in said article of manufacture further comprising computer readable program code means for causing a computer to effect:

employing a proxy between a client and a server to provide connection links between said client and said server;

embedding a secure coprocessor for use as an agent of said client and/or said server;

said coprocessor is located at said proxy site and: (a) acts as a converter between at least one protocol said client supports, and at least one other protocol supported by said server, b) assures that said proxy cannot tamper with the functioning of said agent, and (c) adapts content supplied by said server to fit constraints of said server and connection links;

employing the respective security protocols of said at least one protocol and said at least one other protocol.

Claim 40. (Previously Presented) A computer program product comprising a computer usable medium having computer readable program code means embodied therein for providing secure communication on a network, the computer readable program code means in said computer program product comprising computer readable program code means for causing a

computer to effect:

securely embedding an agent at the site of a proxy in the network, and

splicing a security protocol of a Wireless Applications Protocol suite (WAP) to that of the Internet Protocol (IP) suite.

Claim 41 (Canceled)

Claim 42. (Canceled) A computer program product comprising a computer usable medium having computer readable program code means embodied therein for providing secure communication on a network, the computer readable program code means in said computer program product comprising computer readable program code means for causing a computer to effect:

securely embedding an agent at a proxy in the network, and

splicing a plurality of secure communication protocols of different protocol suites into said agent, wherein said splicing includes maintaining end to end security guarantees at said server.

43. (Canceled) A computer program product comprising a computer usable medium having computer readable program code means embodied therein for providing secure communication on a network, the computer readable program code means in said computer program product further comprising computer readable program code means for causing a computer to effect:

securely embedding an agent at a proxy in the network, and

said agent performing at least one content adaptation function;

splicing a plurality of secure communication protocols of different protocol suites into said agent.

Claim 44. (Canceled) A computer program product comprising a computer usable medium having computer readable program code means embodied therein for providing secure communication on a network, the computer readable program code means in said computer program product further comprising computer readable program code means for causing a computer to effect :

securely embedding an agent at a proxy in the network, and

splicing a plurality of secure communication protocols of different protocol suites into said agent;

maintaining a state of said splicing process resulting from said step of splicing, wherein said step of maintaining includes employing a storage device external to said proxy, and using cryptographic means to encrypt the state of a splicing process resulting from the step of splicing.

Claim 45 (Canceled)

Claim 46 (Canceled) A computer program product comprising a computer usable medium having computer readable program code means embodied therein for providing network security to a network employing a proxy, the computer readable program code means in said computer program product comprising computer readable program code means for causing a computer to effect the steps of :

embedding a trusted application in a secure coprocessor located at the site of a proxy; and

delegating to a network infrastructure a task of enforcing a trust model.

Claim 47 (Canceled) A computer program product as recited in claim 46, the computer readable

program code means in said computer program product further comprising computer readable program code means for causing a computer to effect the step of guaranteeing that the application is trusted to enforce the trust model between at least one server and a plurality of clients.

Claim 48 (Canceled) A computer program product as recited in claim 46, the computer readable program code means in said computer program product further comprising computer readable program code means for causing a computer to effect the step of assuring the tamper resistance of the application.

Claim 49 (Canceled) A program storage device readable by machine, tangibly embodying a program of instructions executable by the machine to perform method steps for secure communication between a client and a server employing an untrusted proxy, said method steps comprising:

embedding a coprocessor at the proxy;
the proxy receiving a specific communication request from a client;
the proxy forming an n-tuple for the specific communication;
the proxy forwarding the n-tuple to the coprocessor;
the coprocessor generating a response, including a directive, to the n-tuple;
the coprocessor sending the response to the proxy, and
the proxy implementing the directive.

Claim 50 (Canceled) A program storage device readable by machine as recited in claim 49, wherein the coprocessor is a secure coprocessor.

Claim 51 (Canceled) A program storage device readable by machine as recited in claim 49, wherein the step of receiving includes:

awaiting a connection request from a first client;
creating an entry in a storage module for the client;
determining a sender of each received packet;

retrieving a stored entry.

Claim 52 (Canceled) A program storage device readable by machine as recited in claim 49, wherein the n-tuple includes a sender id, an entry from a storage module and the received packet.

Claim 53 (Canceled) A program storage device readable by machine as recited in claim 49, wherein the client and the server can be either a sender or a receiver, and the step of generating includes employing a first protocol from the sender to the proxy and a second protocol from the proxy to the receiver and translating between the first and second protocols.

Claim 54 (Canceled) A program storage device readable by machine as recited in claim 49, wherein the translating includes decrypting the received packet as specified by the security parameters negotiated as per the first protocol and encrypting the decrypted packet as specified by the security parameters of the second protocol.

Claim 55 (Canceled) A program storage device readable by machine as recited in claim 49, wherein the translating includes modifying the received packet to meet constraints of the receiver and wherein the directive includes forwarding to the receiver the packet resulting from the step of modifying.

Claim 56 (Canceled) A program storage device readable by machine as recited in claim 55, said method steps further comprising the step of aggregating a plurality of packets into a group of packets and performing content adaptation on the group of packets.

Claim 57 (Canceled) A program storage device readable by machine as recited in claim 49, wherein the communication between the client and the proxy employ protocols specified by the Wireless Application Protocol suite (WAP).

Claim 58 (Canceled) A method as recited in claim 1, further comprising the step of the

coprocessor adapting content supplied by the server to fit constraints of the client and/or the connection links.

Claim 59. (Previously presented) A method as recited in claim 7, wherein the splicing includes maintaining end to end security guarantees without a modification to a server involved in the communication.